exam1-f22

Due Oct 14 at 12:15pm **Points** 100 **Questions** 20
Available Oct 12 at 12am - Oct 14 at 11:59pm **Time Limit** 90 Minutes

Instructions

Pen and paper needed for recursion problems.

Nothing else running except your browser.

This quiz was locked Oct 14 at 11:59pm.

Attempt History

	Attempt	Time	Score
LATEST	**Attempt 1**	62 minutes	21.5 out of 100

Score for this quiz: **21.5** out of 100
Submitted Oct 14 at 11:46am
This attempt took 62 minutes.

Question 1 — 3 / 3 pts

Is ArrayList<Integer> a subclass of ArrayList<? extends Number>?

Correct!

- (•) Yes
- ( ) No

See Figure 19.6.

Question 2

3 / 3 pts

Is ArrayList<Integer> a subclass of ArrayList<?>?

Correct!

- (selected) Yes
- No

See Figure 19.6.

Question 3

0 / 4 pts

Is ArrayList<Integer> a subclass of ArrayList<Object>?

ou Answered

- (selected) Yes

orrect Answer

- No

Question 4

3 / 3 pts

Which of the following statements are true?

- [ ] Generic type information is present at runtime.
- [ ] You can create an array using a generic class type parameter.

Correct!

- [x] Generic type information is present at compile time.

Correct!

- [x] You cannot create an instance using a generic class type parameter.

See the discussion in this section.

Question 5

3 / 3 pts

Is ArrayList<?> same as ArrayList<? extends Number>?

- [ ] Yes

Correct!

- [x] No

? is same as ? extends Object not ? extends Number

Question 6

1.5 / 3 pts

Suppose List list = new ArrayList(). Which of the following operations are correct?

Correct!

- [x] list.add("Red");

orrect Answer

- [ ] list.add(new java.util.Date());

Correct!

- [x] list.add(new Integer(100));

orrect Answer

- [ ] list.add(new ArrayList());

Question 7

0 / 3 pts

Indicate where a compiling error exists.

```
import java.util.ArrayList;
import java.util.List;
public class G<T1>//line 0
{
List<? extends T1> listA = new ArrayList<T1>();//line 1
List<? extends T1> listC = new ArrayList<>(); //line 2
List<? extends T1> listB = new ArrayList<Integer>(); //line 3
}
```

- line 0
- line 1

ou Answered

- line 2 (selected)

orrect Answer

- line 3

Line 3 create a compiling error because we try to assign to type variable T1 a Integer type. Rule 8.

Question 8

0 / 3 pts

Indicate where a compiling error(s) exist(s).

```
public <T2> T2  t2 (T2 t2)//line 0
{
   T2 t1 = new Object();//line 1
```

```
    T2 t3 = new T2();//line 2
    T2 t4 = t2;//line 3
    return t2; //line 4
}
```

- [ ] line 0

orrect Answer

- [ ] line1

orrect Answer

- [ ] line 2

ou Answered

- [x] line 3

- [ ] line 4

public <T2> T2 t2 (T2 t2)//line 0

line 0 compiles: declares <T2>, declares that it returns T2, has a name t2 and takes a parameter a T2 type

T2 t1 = new Object();//line 1

line 1 does not compile, needs a typecast(T2) to compile

T2 t3 = new T2();//line 2

line 2 does not compile , we cannot use the new to create a new generic type

T2 t4 = t2;//line 3

line 3 compiles fine, we use type T2 in both left and right side of the assignment operator

return t2; //line 4

line 3 compiles fine, we are supposed to return T2 and we reutrn the parameter t2 of type T2

Question 9

4 / 4 pts

Indicate where a compiling error exist.

```
ArrayList<? super Number> a = new ArrayList<Number>();//line 1
ArrayList<? extends Number> b = new ArrayList<Number>();//line 2
```

- line 1
- line 2
- line 1 and line 2
- no compiling errors

Correct!

Question 10

4 / 4 pts

Indicate where a compiling error exist.

```
import java.util.ArrayList;
import java.util.List;
public class G<T1>//line 0
{
List<? super T1> listA = new ArrayList<T1>();//line 1
List<? super T1> listB = new ArrayList<?>(); //line 2
List<? super T1> listC = new ArrayList<>(); //line 3
List<?> listD = new ArrayList<Integer>(); //line 4
}
```

- line 0
- line 1
- line 2
- line 3

Correct!

line 4

Question 11

0 / 1 pts

Which line generates a compiling error?

```
package generics.questions;


public class Q6_1
{

}
abstract class Animal1{}
abstract class Plant1{}
class Grass1 extends Plant1{}
interface Hungry1<E> { void eats( E x); }

interface Harnivore1<E extends Plant1> extends Hungry1<E>{}

class Goat1 extends Animal1 implements Harnivore1<Plant1>
   {

     @Override
    public void eats(Plant1 x) { System.out.println("eats"); }

    public static void main(String[] args)
    {
    Goat1 goat1 = new Goat1(); //line 1
     goat1.eats( new Grass1()); //line 2


     goat1.eats( new Plant1() { } ); //line 3

//the braces in bold in line 3 means it has implemented all abstract
methods
          goat1.eats( new Harnivore1<Plant1>()) ;//line 4

  Harnivore1 h1 = new Harnivore1<Plant1>() //line 5
     {
      @Override
        public void eats(Plant1 x){}
   };

 }
}
```

line 1

line 2

line 3

orrect Answer

line 4

ou Answered

line 5

Line 3 doesn't generate an error , we say new to an abstract class Plant1 but the interface has no abstract methods to implement which is denoted by the {} in line 3.

Line 4 generates a compiling error, because we have not implemented its abstract method eats. If we implement them will still generate a compiling error because method east expects a Plant1 not a Harnovore1.

Line 5 is good because we assign a Hanvovore1 to a Harnivore1, in contrast with line 4 where we assign a harnovore1 to a Plant1, via method eats.

Question 12

0 / 4 pts

The declaration compiles:

class B<? extends Object>{}

ou Answered

True

orrect Answer

False

We cannot use wildcard in Type of class declaration.

class B<**T** extends Object>{} would have been correct but not

class B<**?** extends Object>{}

Question 13

0 / 4 pts

Check all that are correct.

```
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;


public class Q7<T>
{
   List<? extends T> list;

   public Q7(ArrayList<? extends T> list) //line 1
     {
    this.list = list;
    System.out.println(this.list);
    }


    public Q7( ArrayList<? super T> list ) //line 2
      {
       this.list = list;
     System.out.println(this.list);
    }

  public static void main(String[] args)
  {
    Q7<CharSequence> q7 = new Q7(
                    new ArrayList<CharSequence>(Arrays.asList("abc")));
//line 3
  }
}
```

- [ ] No compiling errors, output is [abc]

Correct!

- [x] compiling error at line 1, if constructor at line 2 DOES exist
- [ ] compiling error at line 1, if constructor at line 2 does NOT exist

orrect Answer

- [ ] compiling error at line 2

ou Answered

- [x] There will be no compiling errors if we remove the constructor at line 1
- [ ] compiling error at line 3

orrect Answer

- [ ] There will be no compiling errors if we remove the constructor at line 2

If we remove this constructor public Q7(ArrayList<? extends T> list) //line1

and the other constructor remains:

```
public Q7( ArrayList<? super T> list ) //line 2
    {
     this.list = list;
    System.out.println(this.list);
   }
```

it will generate a compiling error in line :

```
this.list = list;
```

by assigning a super of T to a sub of T

Line 3 does not generate compiling errors since in the diamond we have a CharSequence which matches both constructors in the diamond. Outside the diamond of line 3 we have an ArrayList which also a match with ArrayList of both contsructors.

Question 14

0 / 4 pts

Select the line(s) that generate compiling error(s).

```
public class X
{
  1. public static void f1(ArrayList<? super Number> list)
```

```
 {
2. System.out.println( list.size());
3. list.size();
}

public static void main(String[] args)
{
4. f1(new ArrayList<Integer>());
5. f1(new ArrayList<Double>());
}
}
```

- [ ] 1
- [ ] 2
- [ ] 3

orrect Answer

- [ ] 4

orrect Answer

- [ ] 5

ou Answered

- [x] No compiling errors

The parameter expects anything that us a super of Number or Number.

Lines 4 and 5: Integer, Double are not super of Number.

Question 15

0 / 4 pts

Check all lines that DO compile:

```
import java.util.ArrayList;
import java.util.List;
public class G
 {
    public static
```

```
        <E extends CharSequence>//type
        List<? super E>//return type
        doIt(  List<E> numbers )
       {
        return null;
       }

  public static void main()
       {
       ArrayList<String> in1 = null;
       ArrayList<CharSequence> in2 = null;
       List<CharSequence> out1 = null;
       List<?> out2 = null;
       doIt(in2);//line 1
       doIt(in1); //line 2
       out1 = doIt(in2);//line 3
      out1 = doIt(in1);//line 4
      out2 = doIt(in1);//line 5
      }
}
```

Correct!

- [x] line 1

orrect Answer

- [ ] line 2

ou Answered

- [x] line 3

ou Answered

- [x] line 4

orrect Answer

- [ ] line 5

Rule 17.

Question 16

0 / 10 pts

What is the output?

```
public class Recursion
{
public static int f1 ( int x )
 {
 if ( x == 1 )
    return 1;
     System.out.println(x);
 return f1( x-1) - 2 * x -1;
 }
 public static void main(String[] args)
  {
     System.out.println( f1( 5 ) );
   }
}
```

ou Answered -20

orrect Answers -31 (with margin: 0)

Question 17

0 / 10 pts

What is the output?

```
public class Recursion
 {
 public static int f3(int x)
 {
 if (x == 1 || x == 2)
        return 1;
return f3((x + 2) / 2) + f3(x / 2);
 }
 public static void main(String[] args)
 {
 System.out.println( f3(6)  );
 }
}
```

ou Answered 7

orrect Answers 5 (with margin: 0)

Question 18

0 / 10 pts

What is the output?

```
public class Recursion
{
   public static void x( String s )
    {
    if ( s.length() == 0 )
        return;

    System.out.print(s.charAt( s.length()-1 ));
    x(s.substring(0, s.length()-2) );
   }
  public static void main(String[] args)
   {
   x("1234");

  }
}
```

ou Answered 4

orrect Answers 42 (with margin: 0)

Question 19

0 / 10 pts

What does the method return for n = 5 ?

```
public static int f1( int n )
{
   if ( n == 1 || n == 2)
        return 2;
```

```
    int f = f1(n-1) - f1( n-2);
return f;
}
```

ou Answered 1

orrect Answers -2 (with margin: 0)

Question 20

0 / 10 pts

What is the output?

```
public class Recursion
{
static int i = 0, j = 0, k = 0;
public static void f4(int a[][], int b[][], int c[][])
{
    if (i < a.length)
        {
        if (j < b[0].length)
            {
            if (k < a[0].length)
                {
                c[i][j] += a[i][k] * b[k][j];
                k++;
                f4(a, b, c);
                }
            k = 0;
            j++;
            f4(a, b, c);
           }
     j = 0;
     i++;
    f4(a, b, c);
    }
}
public static void main(String[] args)
{
int[][]a = { {1,2}, {3,2},};
```

```
int[][]b = { {3,1}, {4,3} };

int[][]c = { {8,0}, {9,0} };

    f4( a, b, c);
    System.out.println( c[0][1]);
 }

}
```

ou Answered 12

orrect Answers 7 (with margin: 0)

Quiz Score: **21.5** out of 100